

N 9 4 - 2 4 3 9

1993
NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM
MARSHALL SPACE FLIGHT CENTER

THE UNIVERSITY OF ALABAMA OF HUNTSVILLE
ERROR CODING SIMULATIONS

| | |
|--------------------------------|---------------------------------|
| Prepared by: | Viveca K. Noble |
| Academic Rank: | Instructor |
| Institution and Department: | Tuskegee University |
| MSFC Colleague(s): | Bernd Seiler Helen L. Thomas |
| NASA/MSFC: | |
| Office: | Astrionics Laboratory |
| Division: | Computers and Communications |
| Branch: | Flight Data Systems |

Introduction

There are various elements such as radio frequency interference (RFI) which may induce errors in data being transmitted via a satellite communication link. When a transmission is affected by interference or other error-causing elements the transmitted data becomes indecipherable. It becomes necessary to implement techniques to recover from these disturbances. The objective of this research is to develop software which simulates error control circuits and evaluate the performance of these modules in various bit error rate environments. The results of the evaluation provides the engineer with information which helps determine the optimal error control scheme.

The Consultative Committee for Space Data Systems (CCSDS) recommends the use of Reed-Solomon (RS) and Convolutional encoders and Viterbi and RS decoders for error correction (Reference [2]). The use of forward error correction techniques greatly reduces the received signal-to noise needed for a certain desired bit error rate. The use of concatenated coding, e.g. inner convolutional code and outer RS code, provides even greater coding gain. The 16-bit Cyclic Redundancy Check (CRC) code is recommended by CCSDS for error detection (Reference [2]).

Evaluation and Implementation

The initial development phase of the simulator required evaluation of custom error correction software generated for Goddard Space Flight Center (GSFC) to determine what modules were applicable to Marshall Space Flight Center's (MSFC) planned laboratory capabilities as illustrated in Figure 1. A block diagram which illustrates the operation of the GSFC software is shown in Figure 2.

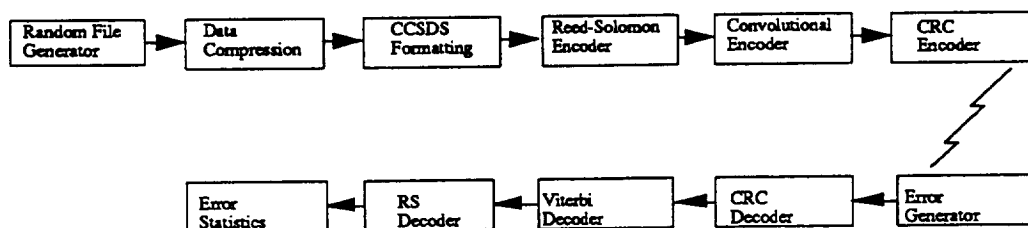


Figure 1



Figure 2

Since the software assumes an all zero input sequence, there is no need for an encoder because the encoded sequence will still be all zeros. This makes the task of determining the error rate a matter of only determining the percentage of non-zero decoder outputs (Reference [5]). Since MSFC's desired system requires random or user-specific data the software from Goddard is unusable in its present form. In order to provide error control capabilities for the Solar X-ray Imager (SXI), the remaining modules of the CCSDS telemetry system simulator were developed. These modules include a multiplicative congruential random number generator (RNG), a random error generator, a CCSDS formatter and a CCSDS recommended CRC error detection encoder/decoder. The error statistics generator is currently being developed.

used to set the 16-bit register to the all 1 state and is given by Equation 3:

$$L(X) = \sum_{i=0}^{15} X_i \quad [3]$$

and $G(X)$ is the generating polynomial given by Equation 4:

$$G(X) = X^{16} + X^{12} + X^5 + 1 \quad [4]$$

The generator polynomial has a Hamming distance of 4 therefore it is guaranteed to detect error sequences composed of one, two or three bit errors (Reference [4]). When this code is applied to a block of less than 32768 (2^{15}) bits, it also has the capability to detect all odd number of bit errors, to detect at most two bit errors, to detect all single burst errors with a length of 16 bits or less as long as there are no other errors in the block and has an undetected error probability of 2^{-15} (or 3×10^{-5}) for a random error sequence containing an even number of bit errors greater than or equal to 4.

The error detection decoder module is the software implementation of Figure 5 (Reference [2]).

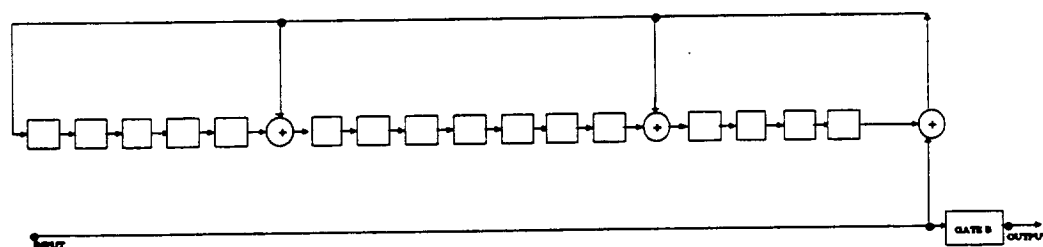


Figure 5

Equation 5 gives the error detection syndrome.

$$S(X) = [X^{16} \cdot C^*(X) + X^n \cdot L(X)] \text{ modulo } G(X) \quad [5]$$

where $C^*(X)$ is the received block in polynomial form and $S(X)$ is the syndrome polynomial. The 16-bit register will contain all zeros if no error is detected and will contain non-zero values if an error is detected. The decoder also attempts to establish synchronization, but if a sync marker error occurs, a message will be generated to indicate this occurrence and zeros will appear in the syndrome polynomial to reflect this error.

The decoder's performance has been verified for up to 3 random errors. Tests will be performed to verify the additional performance characteristics. In generating statistics on the error detection capability, various bit error rate environments will be created and decoded for a number of successive runs. The error statistics generator will assign a one for each non-zero syndrome and a zero for each zero syndrome. It will determine the error statistics based on the percentage of non-zero terms.

Conclusion and Future Tasks

All of the previously discussed software is written in FORTRAN 77. Due to the inflexible nature of this language, e.g. input data arrays must be given a declared size, it is recommended that the code be converted to C and all future code be written in C. Appropriate error distributions must be determined so that customized error control environments may be developed. The current error correction portion of the system must be written for use with random data and user specific data. Convolutional and RS encoders and a more refined and flexible error generator must be developed. Data compression modules need to be added for the handling of "housekeeping" data. Testing of the code for various bit error rates must be continued in order to gather statistical data on the performance of the code. The process presented above provides a modular, inexpensive error control environment. Its use will allow an engineer to create an optimal error control environment for a given error distribution prior to implementing the procedure in hardware.

References

- [1] *Telemetry Channel Coding*, Recommendation CCSDS 101.0-B-3, Issue 3, Blue Book, Consultative Committee for Space Data Systems, May 1992 or later issue, p. 5- 1.
- [2] *Telemetry*, Recommendation CCSDS 100.0-G-1, Issue 1, Green Book, Consultative Committee for Space Data Systems, December 1987 or later issue, pp. 3-19 - 3-20 and pp. D-1 - D-4.
- [3] Hamming, R. W., *Numerical Methods for Scientists and Engineers*, Dover Publications, Inc., New York, 1986
- [4] Jain, Raj, *Error Characteristics of Fiber Distributed Data Interface (FDDI)*, IEEE Transaction on Communications, Vol. 38, No. 8, August 1990, p. 1249.
- [5] Odenwalder, Joseph P., *Error Control Coding Handbook*, Final Report, Contract F44620-76-C-0056, July 15, 1976, p. 4 and p. 122.